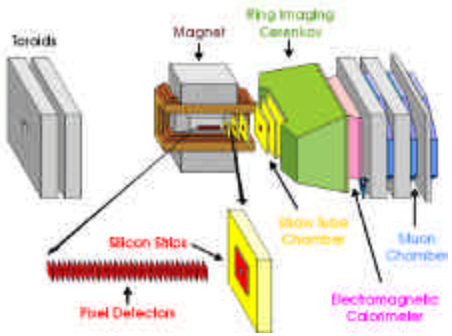


The BTeV experiment is designed to challenge the Standard Model explanation of CP violation, mixing and rare decays of beauty and charm quark states. The BTeV Collaboration is a group of about 170 physicists. The experiment will utilize the Tevatron proton-antiproton collider at the Fermi National Accelerator Lab.

BTEV Co

Contact Information

Jim Kowalkowski
Joel Butler

Paul Sheldon
Ted Bapty
Sandeep Neema

Daniel Mosse

Ravikant Iyer
Michael Haney
Zbigniew Kalbarczyk

Jae Oh

For more information on the ongoing research and collaborating groups please visit the following web-sites:

General information about RTES:

http://www-rtev.fnal.gov/public/hep/rtes/

General information about BTeV:

http://www-btev.fnal.gov/public/gen/index.shtml

Information about the Vanderbilt research group:

http://www.isis.vanderbilt.edu/btev

http://www.isis.vanderbilt.edu/view.asp?GID=120&CAT=3

~~/rtes~~

Information about ARMOR technology:

http://www.crhc.uiuc.edu/DEPEND/rtes.htm

Talks from last RTES workshop:

http://false2002.vanderbilt.edu/program.php/

RTES

Real-Time, Embedded Systems group

Illinois Vanderbilt Pittsburgh ~~Vanderbilt~~ Syracuse Fermilab

Super Computing 2003

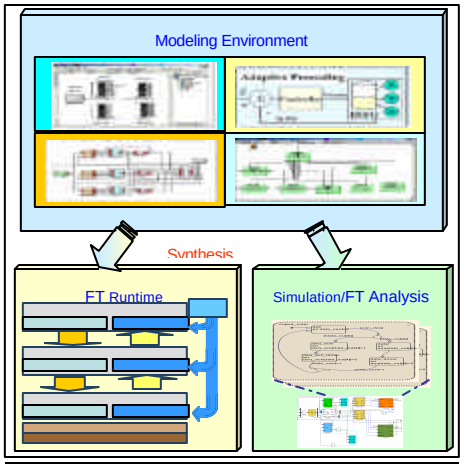
Phoenix, AZ

November 15th-21st 2003

Physicists, Computer Scientists, and Electrical Engineers with expertise in high performance, real time, embedded system software and hardware, reliability, and fault tolerance, system specification, generation, and modeling tools, doing research on fault management in large computing clusters with real time needs.

Funded by NSF grant ACI-0121658

Modeling and Generation tools for Fault Adaptive, Real Time, Large Scale, Embedded Systems



Models define:

- System Hardware
- Application Structure
- Fault Behavior
- Reconfiguration

Automatic Synthesis of:

- Real-Time Schedules
- Kernel Configuration
- Communication Maps
- System Managers
- Reconfiguration Reflex and Healing Actions

Simulation Framework

- Behavioral models of Hardware-software Components
- System simulation automatically composed by model-synthesis tools

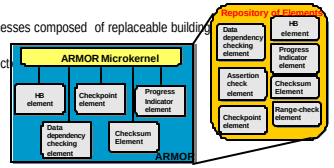
Run-Time Environment:

- Fault Adaptive Real-Time Kernel
- Hardware-supported Fault Recovery
- Redundant Communication
- Interface to MIC Synthesizer

ARMOR-based Hierarchical Error Management

What are ARMORs?

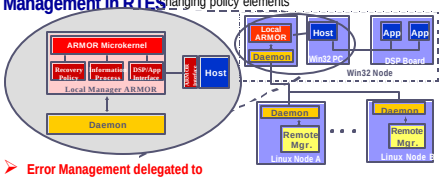
- Multithreaded processes composed of replaceable building blocks (elements)
- Provide error detection and recovery services to user applications
- Hierarchy of ARMORs forms self-checking runtime environment
- System management, error detection, and recovery services distributed across ARMOR processes.



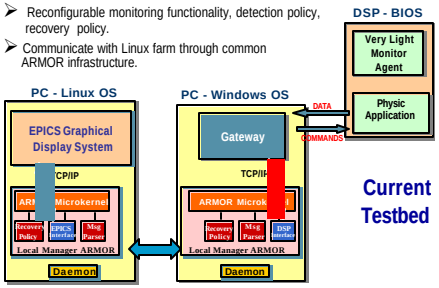
Computational Model

- Elements invoked through operations executing within a tread.
- Element can: read/write variables and element state, or generate new operations
- **Element-based detection and recovery:**
  - Monitor generates operation on an error.
  - Policy elements generate sequence of operations to effect recovery.
- Response to errors can be reconfigured by changing policy elements

ARMOR-Based Fault Management in RTES



- **Error Management delegated to ARMOR processes:**
- Reconfigurable monitoring functionality, detection policy, recovery policy.
- Communicate with Linux farm through common ARMOR infrastructure.



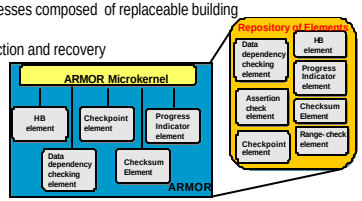
Current Testbed

Implementation

ARMOR-based Hierarchical Error Management

What are ARMORs?

- Multithreaded processes composed of replaceable building blocks (elements)
- Provide error detection and recovery services to user applications
- Hierarchy of ARMORs forms self-checking runtime environment
- System management, error detection, and recovery services distributed across ARMOR processes.

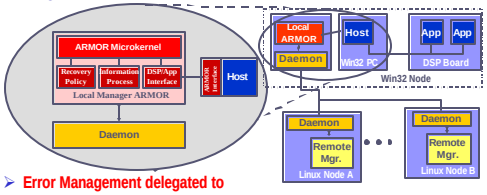


Computational Model

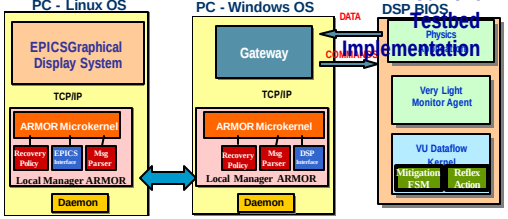
- Elements invoked through operations executing within a tread.
- Element can: read/write variables and element state, or generate new operations
- **Element-based detection and recovery:**
  - Monitor generates operation on an error.
  - Policy elements generate sequence of operations to effect recovery.
- Response to errors can be reconfigured by changing policy elements

ARMOR-Based Fault Management in RTES

ARMOR-Based Fault Management in RTES



- **Error Management delegated to ARMOR processes:**
- Reconfigurable monitoring functionality, detection policy, recovery policy.
- Communicate with Linux farm through common ARMOR infrastructure.



Current Testbed Implementation

Current Testbed Implementation

Very Lightweight **Agents (VLA)**

- Monitor hardware integrity
- Monitor software integrity
- Intelligent and Adaptive (e.g., error prediction, correction)
- Reactive, Proactive, Cooperative
- Small Footprint
- Hierarchical